ซอฟต์แวร์
รัจพันธ์ ปติฐพร

น้องโบเขียนไทย ไม่พึ่งฮาร์ดแวร์

การออกแบบตัวฟอนต์ ใช้กับจอ CGA ในโหมดกราฟิก

ผมต้องบอกก่อนว่าน้องโบในที่นี้ไม่ใช่ใครอื่น หมายถึงเทอร์โบปาสคาล ที่มาแรงไม่เคยโดนใครแซงหน้าอยู่ในขณะนี้ไงครับ

คือมีอยู่ว่าผมเป็นคนหนึ่งที่มีคอมพิวเตอร์ไอบีเอ็มคอมแพตทิเบิ้ล CGA โปรแกรมพวกเวิร์ดโปรเซสเซอร์ที่เขียนด้วยกราฟิกที่ไม่พึ่งฮาร์ดแวร์ก็มักจะไม่มีให้สำหรับจอ CGA ผมจึงพยายามเขียนโปรแกรมนี้ขึ้นมาเพื่อใช้งานและนำเสนอมาให้ผู้อ่านลองไปใช้ดูบ้างโปรแกรมนี้ต้องการใช้ฟอนต์ของ CU Writer เพราะผมเห็นว่า CU Writer มีอยู่ทั่วไปและสะดวกที่ไม่ต้องเขียนฟอนต์เอดิเตอร์เองสำหรับท่านผู้อ่านก็ก๊อปปี้เอาไฟล์ "NORMAL.FON" มาไว้ในแผ่นที่ Run โปรแกรม ก็สามารถจะใช้โปรแกรมยูนิตของผมได้เลย เพราะฉะนั้นโปรแกรมยูนิตของผมได้เลย เพราะฉะนั้นโปรแกรมนี้จึงมีไว้สำหรับท่านที่มีจอสี CGA และจอโมโนโครมเฮอร์คิวลิสครับ แต่คิดว่าคุณผู้อ่านที่มีจอ EGA คงนำยูนิตนี้ไปเพิ่มเติมหรือดัดแปลงใช้เองได้ไม่ยากครับ

โปรแกรมนี้เขียนด้วยเทอร์โบปาสคาลเวอร์ชัน 5.0 แต่เวอร์ชัน 4.0 ก็ใช้ได้เหมือนกัน เพราะผมใช้แต่คู่มือของเวอร์ชัน 4.0 เป็นหนังสืออ้างอิง โปรแกรมนี้ทำเป็นยูนิตชื่อว่า THAIGR .PAS พอคอมพายล์แล้วกลายเป็น THAIGR.TPU แต่ตอนเรียกใช้ให้เรียก THAIGR ก่อน การใช้ฟังก์ชันต่าง ๆ ในยูนิตนี้มีโปรแกรมตัวอย่างให้มาด้วยแล้ว ฟังก์ชันต่าง ๆ ในยูนิตจะเลียนแบบฟังก์ชันเอาต์พุตออกจอภาพในโหมดการแสดงผลตัวอักษร (Text mode) ในภาษาปาสคาล ตัวอย่างเช่น กำหนดตำแหน่งรอการพิมพ์ (GotoXY), พิมพ์ตัวอักษรเป็นบรรทัด (WriteLn), ลบจอภาพ (ClrScr) รับคีย์ตัวอักษรเป็นบรรทัด (ReadLn) เป็นต้น ส่วนในด้านรูปแบบของตัวอักษรโดยปกติในเท็กซ์โหมดทั่ว ๆ ไป จะถูกกำหนดโดยแอตทริบิวต์ให้เกิดเป็นตัวเข้ม, ตัวกะ-

พริบ หรือ ตัวสีต่าง ๆ สำหรับจอสี แต่ในยูนิต THAIGR ไม่มีเพราะเป็นโหมดกราฟิก ทำได้ก็แต่รูปแบบฟอนต์ต่าง ๆ คือ ตัวอินเวิร์สและตัวหนา (Bold) ส่วนตัวอื่น ๆ เช่นตัวเอียง (Italic) ตัวใหญ่ (Enlarge) ผมไม่ได้ทำไว้ เพราะคิดว่าคงเพียงพอและเพื่อไม่ให้โปรแกรมทำงานช้าเกินไปด้วยครับ

หลักการ

รายละเอียดของหลักการเขียนตัวอักษรไทยออกจอภาพกราฟิก ก็คือ

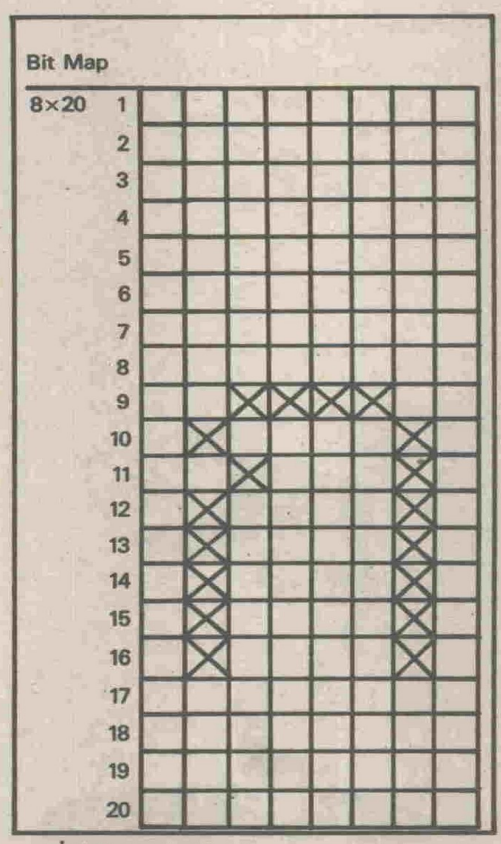
1. ออกแบบตัวฟอนต์และเก็บไว้ที่ไหนสักแห่ง เช่น ในไฟล์ที่อยู่คนละส่วนกับโปรแกรมหรือไม่ก็เก็บไว้เป็นค่าคงที่ในโปรแกรมเลย ท่านผู้อ่านนำไฟล์ที่ชื่อว่า "NORMAL.FON" ของ CU Writer มารวมไว้ในแผ่นดิสค์ที่มียูนิต THAIGR.PAS

2. เขียนโพรซีเดอร์หรือฟังก์ชันจัดการอ่านฟอนต์มาเก็บไว้ในตัวแปร Array เมื่อเรียกใช้ก็ใส่ข้อมูลของฟอนต์ซึ่งเก็บไว้เป็นไบต์ไปไว้ยังเมมโมรีโลเคชันของการ์ดแสดงผลนั้น ๆ โดยใช้คำสั่งประเภท Poke (ถ้าเป็นภาษาเบสิก) หรือ Mem (ในเทอร์โบปาสคาล) และต้องคำนวณตำแหน่งให้เกิดรูปร่างตัวอักษรที่ถูกต้อง เมื่ออยู่ในโหมดกราฟิกถึงจะเกิดเป็นรูปร่างขึ้นมา ส่วนการทำตัวอักษรให้เกิดรูปฟอนต์ต่าง ๆ นั้นอาศัยการเปลี่ยนแปลงข้อมูลที่เป็นไบต์นั้น ๆ ให้เป็นตัวหนา หรือตัวอินเวอร์สก่อนที่จะมีการนำข้อมูลไปไว้ที่เมมโมรีโลเคชันของการ์ดแสดงผล

3. ใช้รหัสภาษาไทยซึ่งมีอยู่หลายแบบ แต่ที่นิยมมากคือสมอ.และเกษตร แต่ผมจะใช้ของสมอ. เพราะเป็นรหัสมาตรฐานของสมาคมมาตรฐานอุตสาหกรรม ซึ่งรหัสตัวอักษรไทยคือไม่มีตัวสระผสม และยังมีตัวบาลีสันสกฤตครบถ้วน เมื่อเลือกรหัสได้แล้วก็จัดการเทียบรหัสภาษาไทยกับตำแหน่งแป้นพิมพ์ ดังนั้นส่วนนี้ก็คือส่วนของฟังก์ชันอินพุตนั่นเอง

เรื่องของฟอนต์

ฟอนต์ของ CU Writer ประกอบด้วยจุดตามแนวคอลัมน์ 8 จุด ตามแถว 20 จุด กราฟิกดังรูปที่ 1 ในแต่ละแถวที่มี 8 จุดซึ่งแทนข้อมูลเลขฐานสอง เช่น ให้จุดของกราฟิก (Pixel) จุดสว่างหนึ่งจุดเป็นเลข 1 บนฐานสอง, จุดมืดหนึ่งจุดเป็นเลข 0 บนฐานสอง ตัวอย่างเช่น ถ้าบนแถวแรกมีจุดมืดอยู่ทางซ้ายสี่จุด และถัดมาอีกสี่จุดเป็นจุดสว่างทั้งหมดก็จะเขียนเป็นเลขฐานสองจำนวน 8 บิตได้เป็น 00001111 อ่านว่า

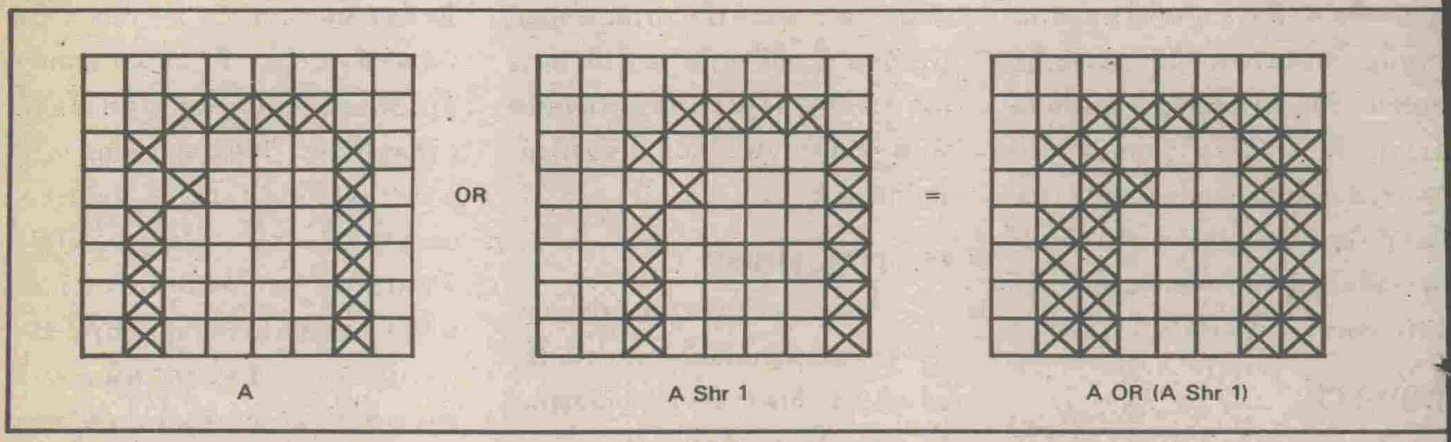


รูปที่ 1 ตัวอย่างฟอนต์ของ CU Writer

บิตที่ตำแหน่ง 0,1,2,3 มีค่าเป็น 1 บิตที่ตำแหน่ง 4,5,6,7 มีค่าเป็น 0 จากเลขฐานสองแปลงเป็นเลขฐานสิบได้ดังนี้ (เรียงจากบิต 7 ไปยังบิต 0 จากซ้ายไปขวา) $0\times(2^7) + 0\times(2^6) + 0\times(2^5) + 0\times(2^4) + 1\times(2^3) + 1(2^2) + 1\times(2^1) + 1\times(2^0) = 15$ ดังนั้นถ้าทั้ง 8 บิตมีค่าเป็น 1 จะแปลงเป็นเลขฐานสิบได้ 255 การเก็บเลข 8 บิตในคอมพิวเตอร์สามารถจะเอามาแทนด้วยค่าของตัวเลขฐานสิบ และสามารถแทนด้วยข้อมูลเป็น Char มีค่าตามตำแหน่งของแอสกีโค้ดจาก 0 ถึง 255 ดังนั้นฟอนต์ที่มีแถว 20 แถวจะใช้เก็บด้วย 20 ตัวอักขระ (หรือ 20 ไบต์) ซึ่งก็คือฟอนต์หนึ่งตัวจะเก็บลงไฟล์ด้วย 20 ตัว, Char ใน CU Writer จะเก็บฟอนต์ทั้งหมด 256 ตัวตามลำดับไว้ในไฟล์ "NORMAL.FON" ดังนั้นขนาดไฟล์จึงเท่ากับ $256\times20 = 5120$ ไบต์

การทำตัวอินเวิร์ส ทำได้ง่ายโดยเอาค่าของไบต์ที่มีค่าสูงสุด (มีจุดสว่างทุกบิต) หักออกด้วยหนึ่งไบต์ของฟอนต์นั้น ๆ นั่นคือ = (255 – ข้อมูลหนึ่งไบต์ของฟอนต์นั้น ๆ)

การทำตัวหนา เนื่องจากฟอนต์ของ CU Writer มีช่องไฟสำหรับตัวอักษรอยู่ทางขวา แต่บางตัวไม่มีที่พอ เช่น ตัว ฌ ญ ฑ ฒ ฌ การทำตัวหนาก็อาจขาดความสวยงามลงไปบ้าง เพราะตัวหนาต้องการเนื้อที่อีกอย่างน้อย 1 จุด หลักการมีอยู่ว่าตัวหนาของผมในที่นี้หมายถึงว่าหนาไม่สมส่วน เพราะจะหนาในทางกว้างเหมือนคนอ้วนเอา ๆ แต่ไม่สูงขึ้น วิธีการก็คือทำฟอนต์ที่เหมือนเดิมทุกประการขึ้นมา แต่ต่างตรงที่ฟอนต์ตัวนี้จะมีทุกจุดเลื่อนตำแหน่งไปอยู่ทางขวาของฟอนต์ตัวเดิมเป็น



รูปที่ 2 ฟอนต์สองตัวทับกันได้เป็นฟอนต์ตัวหนา

ระยะทางหนึ่งจุด ดังรูปที่ 2 ถ้าฟอนต์สองตัวนี้นำมาทับกันจะเกิดอะไรขึ้น สิ่งที่เกิดขึ้นคือจะเห็นฟอนต์ตัวหนากว่าเดิมไงครับ การเขียนโปรแกรมให้เกิดการเลื่อนของทุก ๆ จุดไปทางขวาหรือซ้ายจะใช้ Shift bit operators ซึ่งมีใช้ในภาษาแอสเซมบลี, ภาษาซี และภาษาปาสคาล สำหรับเทอร์โบปาสคาลมีคำสั่งว่า shr (shift right operator) และ shl (shift left operator) ตัวอย่างเช่นใช้คำสั่งว่า A : = B shr 1; หมายความว่าให้เลื่อนค่าที่อยู่ประจำตำแหน่งบิตเดิมของทุก ๆ บิตของค่าที่ลอกมาจาก B ไปอยู่ตำแหน่งถัดมาทางขวาหนึ่งตำแหน่ง เช่น 11110000 จะได้เป็น 01111000 สังเกตว่าตำแหน่งบิตที่ 7 จะมีค่า 0 มาแทนที่ ไม่ได้เกิดจากค่าในบิตที่ 0 ไปแทนที่ แต่มันจะเป็นเช่นนี้ใสมอคือเอา 0 มาทดให้แก่ปลายด้านที่เลื่อนจากออกมา มันเป็นคุณสมบัติของตัวโอเปเรเตอร์เอง แล้ว A ก็จะมีค่าเป็นค่าของ B ที่ถูกเลื่อนบิตแล้ว ส่วนการเขียนให้เกิดการทับกันของฟอนต์ ทำได้โดยการรวมกันของบิต ไม่ใช่การรวมกันแบบบวกเลขธรรมดา แต่ต้องรวมแบบบิตโดยการนำมาทำ Bitwise OR ระหว่างข้อมูลทั้งสอง (ข้อมูลตัวแปรอาจเป็น Integer, Byte, Word หรือ Char สำหรับ Char ในภาษาซีใช้ได้แต่ในปาสคาลใช้ไม่ได้ ในที่นี้ผมใช้ข้อมูลเป็น Byte)

ตัวอย่างฟังก์ชันแปลงฟอนต์เป็นตัวหนา

```
Function Bold (a : byte) : byte ;
var b : byte ;
begin
   b:=a shr 1;
   Bold : = a or b ;
```

เรื่องของจอภาพกราฟิก

จอภาพ CGA

จอภาพ CGA แสดงจุดได้ 640 จุดตามแนวนอน 200 จุดตามแนวตั้ง เพื่อความเข้าใจอย่างง่าย ๆ ของให้มองจอภาพกราฟิกถูกตีเป็นเส้นตามแนวนอนด้วยจำนวนจุดทั้งหมด 640 จุด ดังนั้นถ้าตีเส้นให้เต็มจอก็จะตีเส้นตามแนวนอนได้ 200 เส้น ถ้าคุณต้องการที่จะวาดเส้นตามแนวนอนเส้นบนสุดของจอภาพ อาจจะทำได้โดยใช้คำสั่ง PutPixel ให้เกิดจุดตำแหน่ง (x,y) (0,0),(1,0),(2,0),(...,0) เรื่อยไปจนถึง (639,0) หรือไม่ก็ใช้คำสั่ง Line (0,0, 639,0) แต่ก็ยังมีอีกวิธีหนึ่งคือใช้คำสั่ง Mem [segment,offset] เช่น Mem [$B800:$2000]:=$FF หมายความว่านำค่า $FF ขนาด 1 ไบต์ไปบรรจุไว้ที่แอดเดรส $B800:$2000 ของหน่วยความจำแสดงผลของจอภาพเมื่ออยู่ในโหมดกราฟิกได้เช่นกัน และถ้าต้องการสร้างเส้นตรงดังกล่าวจะต้องบรรจุค่า $FF (เทียบเท่า 255 ฐานสิบ หรือ 11111111 ฐานสอง ซึ่งเป็นจุดสว่างหมด) ไปที่เซกเมนต์ $B800,Offset $0000 จำนวน 80 ตัวก็จะได้เส้นตรงที่

ต้องการได้เหมือนกัน แต่จะเร็วกว่า เพราะเป็นการป้อนค่าสู่หน่วยความจำโดยตรง ไม่ต้องผ่านฟังก์ชันที่ซับซ้อน ซึ่งจะทำงานช้ากว่า ถ้าบรรจุตัวที่ 81 ต่อไปจนครบ 160 ตัวจะได้เป็นเส้นแนวนอน แต่เป็นเส้นที่ 3 ข้ามเส้นที่ 2 ไปแล้วก็จะเป็นเช่นนี้ร่ำไปเมื่อขึ้นเส้นใหม่ หมายถึงได้เส้นที่ 3,5,7,...,199 (เส้นคี่) เป็นเพราะเส้นที่สองเริ่มต้นที่แอดเดรส $B800;$2000 และเช่นเดียวกันเมื่อบรรจุครบ 80 ตัวที่แอดเดรส $B800:$2000 จะได้เส้นที่สองตามแนวนอนพอบรรจุตัวที่ 81 เป็นต้นไปและต่อ ๆ ไปก็จะได้เส้นที่ 4,6,8,...,200 (เส้นคู่)

ส่วนฟอนต์ประกอบด้วยข้อมูล 20 ไบต์แต่ละไบต์แสดงหนึ่งแถว ถ้าบรรจุลงไปยังหน่วยความจำแสดงผลจอภาพกราฟิกตามเส้นแนวนอนตั้งแต่เส้นที่ 1 ถึง 20 และให้คอลัมน์ตรงกันก็จะได้ฟอนต์ตัวนั้นปรากฏอยู่บนจอภาพกราฟิกได้

จอภาพโมโนโครมกราฟิก

สำหรับจอภาพโมโนโครมกราฟิกก็อาศัยหลักการเดียวกัน แต่มีความแตกต่างกันบางประการคือ จอภาพแสดงผลได้จำนวนจุดมากกว่าคือ 720 × 348 จุด ดังนั้นเส้นหนึ่ง ๆ ตามแนวนอนเมื่อบรรจุข้อมูลแก่หน่วยความจำจอภาพได้ 90 ไบต์ก็จะได้เส้นตามแนวนอนหนึ่งเส้น และที่แตกต่างอีกอย่างคือเส้นตามแนวนอนเส้นที่ 1,2,3,4 เริ่มที่แอดเดรส $B000:$0000, $B000:$2000, $B000;$4000, $B000:$6000 ตามลำดับ จะมีการกระโดดของเส้นตามแนวนอนอย่างนี้เป็นช่วง ๆ เป็นจังหวะ ๆ ไป

เกี่ยวกับรหัสภาษาไทย

ใช้รหัสของสมอ.เพื่อความเป็นมาตรฐาน สำหรับคีย์บอร์ดภาษาไทยของคุณอาจจะไม่ตรงกับตำแหน่งคีย์อินพุตในฟังก์ชัน GrReadLn คุณอาจแก้ไขเปลี่ยนแปลงตำแหน่งคีย์บอร์ดไทยเทียบกับคีย์บอร์ดภาษาอังกฤษของคุณเอง โดยศึกษาตัวแปล ThCode [] ที่ declare ไว้ต้นยูนิต THAIGR

มีอะไรในยูนิต THAIGR

เนื่องจากฟอนต์มีขนาด 8×20 pixels ดังนั้นจอ CGA 640×200 จะแสดงผลภาษาไทยได้สูงสุด 80 คอลัมน์ (640/8) แถวสูงสุด 10 แถว (200/20) ส่วนจอโมโนโครมกราฟิกมีขนาด 720 ×348 จุด จึงแสดงผลภาษาไทยได้สูงสุด 90 คอลัมน์ 17 แถว (มีเศษเหลือ 8 pixels ตามแนวนอน) ในยูนิต THAIGR มี Global variable GrMaxColumn และ GrMaxRow จะแสดงค่าสูงสุดของคอลัมน์และแถวสำหรับจอภาพนั้น ๆ

Global variables ที่ให้ผู้ใช้เรียกใช้ (สังเกตได้จากโปรแกรม ตัวแปรจะไม่มีเครื่องหมาย "-" นำหน้า) ได้แก่

- GrMaxColumn, GrMaxRow เป็นอินทีเจอร์ให้เรียกใช้ได้แต่ห้ามเปลี่ยนแปลง

- InverseForm, BoldForm เป็นบูลีน ถ้าให้ค่านี้เท่ากับ True จะทำให้เกิดตัวอินเวิร์สและตัวหนาตามลำดับ

Procedure และ Function

- procedure InitGrThaiIO;
- procedure CloseGrThaiIO;

ก่อนการเรียกใช้ฟังก์ชันต่าง ๆ ทุกครั้งต้องเรียก InitGrThaiIO; เสมอ เพื่อเข้าสู่โหมดกราฟิกและเซ็ตค่าเริ่มต้นต่าง ๆ และเมื่อจบการใช้งานจะต้องเรียก CloseGrThaiIO; เพื่อเปลี่ยนกลับไปสู่เท็กซ์โหมดและคืนหน่วยความจำบางส่วนให้แก่ระบบ (Deallocate memory)

- procedure GrWindow (x1, y1, x2, y2 : integer);

กำหนดขอบเขตหน้าต่างจอภาพที่ต้องการใช้งาน โดย x1, y1 คือตำแหน่งพิกัดมุมซ้ายบน x2, y2 คือตำแหน่งพิกัดมุมล่างขวา x มีค่าระหว่าง 1...GrMaxColumn และ y มีค่าระหว่าง 1...GrMaxRow ถ้าผู้ใช้กำหนดค่านอกขอบเขตนี้โปรแกรมจะติดตั้งค่าสูงสุดให้

- procedure GrGotoXY(x, y : integer);

กำหนดตำแหน่งรอการพิมพ์ที่พิกัด x,y ถ้ากำหนดค่าที่ไม่อยู่ในขอบเขตของหน้าต่าง ตำแหน่งรอการพิมพ์จะถูกกำหนดให้ไปอยู่ที่มุมซ้ายบนของหน้าต่างที่กำลังใช้งานอยู่

- precedure GrClrScr;

ลบจอภาพแล้วย้ายตำแหน่งรอการพิมพ์ไปที่มุมซ้ายบนของหน้าต่างที่กำลังใช้งานอยู่

- procedure GrScrollUp;

เลื่อนจอภาพที่อยู่ในขอบเขตของหน้าต่างขณะนั้นขึ้นหนึ่งบรรทัด

- procedure GrScrollDown;

เลื่อนจอภาพที่อยู่ในขอบเขตหน้าต่างนั้นลงหนึ่งบรรทัด

- procedure GrWriteCh (c: char);

เขียนตัวอักขระออกจอภาพหนึ่งตัว

- procedure GrWrite (__st: str255);

เขียนตัวอักขระเป็นบรรทัดจนจบบรรทัด แล้วไม่เลื่อนตำแหน่งรอการพิมพ์ไปบรรทัดใหม่

- procedure GrWriteLn (__st :str255);

เขียนตัวอักษรเป็นบรรทัดจนจบบรรทัดแล้วเลื่อนตำแหน่งรอการพิมพ์ไปบรรทัดใหม่

- function GrReadLn:str255;

รอรับข้อมูลจากคีย์บอร์ดเป็นบรรทัดจนกว่าจะกดปุ่ม ENTER ค่าสูงสุดของคีย์บอร์ดบัฟเฟอร์คือ 255 ตัวอักขระ การเปลี่ยนการใช้คีย์บอร์ดเป็นไทยหรืออังกฤษทำได้โดยกดปุ่ม F1(จะเปลี่ยนกลับไปกลับมาระหว่างแป้นพิมพ์ไทยและอังกฤษ การกำหนด Boolean __ThaiMode ให้เป็น True หรือ False จะเป็นการตั้งแป้นพิมพ์เริ่มแรกให้เป็นไทยหรืออังกฤษตามลำดับ

เครียดกับรายละเอียดกันมามากพอสมควร เพื่อให้สมบูรณ์แบบและเพื่อให้เข้าใจง่ายผมมีตัวอย่างการใช้ฟังก์ชันต่าง ๆ (ดูรูปที่ 3) มาให้คุณผู้อ่านที่รักด้วย ลองศึกษาดูสิครับ.

```
Program Thdemo;
uses Crt,ThaiGr;

var
    i  : integer;
    ch : char;
    st : str255;
    f  : text;
BEGIN
   InitGrThaiIO; { Using ThaiGr unit requires the initialize rutine }
   assign(f,'thaitxt.dat');
   reset(f);
 while not eof(f) do begin
   readln(f,st);
   GrWriteLn(st);
 end;
 ch := readkey;
   close(f);
   InverseForm:=True;
   GrClrScr;
   GrGotoXY(25,1); GrWriteLn('[ This is the example programs.]');
   GrWindow(15,2,65,9);
   InverseForm:=False;
   GrClrScr;
   st:=GrReadLn;
   GrWrite(st);ch:=readkey;GrWriteLn(st);ch:=ReadKey;
```

รูปที่ 3 โปรแกรมตัวอย่างที่เรียกใช้ฟังก์ชันในยูนิต THAIGR

รูปที่ 3 (ต่อ)

```
GrClrScr;
GrWriteLn('< This is normal character fonts.>');
for i:=1 to 256 do  GrWriteCh(chr(i));
ch := readkey;

GrClrScr;
GrWriteLn('< This is bold character fonts.>');
BoldForm:=True;
for i:=1 to 256 do  GrWriteCh(chr(i));
ch := readkey;

GrClrScr;InverseForm:=True;
GrWriteLn('< This is inverse character fonts.>');
for i:=1 to 256 do  GrWriteCh(chr(i));
BoldForm:=False;
for i:=1 to 256 do  GrWriteCh(chr(i));
ch := readkey;

BoldForm:=True;InverseForm:=False;
GrScrollDown; GrGotoXY(15,2);GrWriteLn('< Scroll down one line >');
ch:= readkey;
GrScrollUp;   GrGotoXY(15,9);GrWriteLn('< Scroll up one line >');
ch:=.readkey;

CloseGrThaiIO; { Close the graphic screen and deallocate memory. }
assign(f,'thaitxt.dat');
append(f);
writeln(f,st);
close(f);
END.
```

คุณใช้งานดีเบส
เต็มประสิทธิภาพหรือยัง?

แอดวานซ์ดีเบส

เป็นหนังสือที่รวบรวมเทคนิคการประยุกต์ใช้งานดีเบสทรีพลัสโดยผู้มีประสบการณ์จากการใช้งานและออกแบบวางระบบงานด้วยดีเบสมาเป็นเวลานาน ยากที่จะหาอ่านได้จากหนังสือเล่มอื่น ภายในเล่มประกอบด้วยเทคนิคต่าง ๆ พร้อมโปรแกรมตัวอย่างประกอบทุกบท เหมาะสำหรับผู้ที่ต้องการนำโปรแกรมดีเบสทรีพลัส มาใช้งานให้เกิดประโยชน์สูงสุดอย่างแท้จริง

หาซื้อได้
ตามร้านหนังสือทั่วไป